Knowl Inf Syst (2017) 53:551–577
DOI 10.1007/s10115-017-1059-8

SURVEY PAPER

Recent advances in feature selection and its applications

Yun Li[1,2] · **Tao Li**[1,2,3] · **Huan Liu**[4]

Received: 19 June 2016 / Accepted: 27 April 2017 / Published online: 5 May 2017

Abstract Feature selection is one of the key problems for machine learning and data mining. In this review paper, a brief historical background of the field is given, followed by a selection of challenges which are of particular current interests, such as feature selection for high-dimensional small sample size data, large-scale data, and secure feature selection. Along with these challenges, some hot topics for feature selection have emerged, e.g., stable feature selection, multi-view feature selection, distributed feature selection, multi-label feature selection, online feature selection, and adversarial feature selection. Then, the recent advances of these topics are surveyed in this paper. For each topic, the existing problems are analyzed, and then, current solutions to these problems are presented and discussed. Besides the topics, some representative applications of feature selection are also introduced, such as applications in bioinformatics, social media, and multimedia retrieval.

Keywords Feature selection · Survey · Data mining

✉ Yun Li
liyun@njupt.edu.cn

Tao Li
towerlee@njupt.edu.cn

Huan Liu
huan.liu@asu.edu

[1] School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing, China

[2] Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, China

[3] School of Computer Science, Florida International University, Miami, FL, USA

[4] School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA

1 Introduction

Feature selection is an important and frequently used technique for dimension reduction by removing irrelevant and redundant information from the data set to obtain an optimal feature subset [1,2]. It is also a knowledge discovery tool for providing insights into the problems through the interpretation of the most relevant features. Feature selection research dates back to the 1960s. Hughes used a general parametric model to study the accuracy of a Bayesian classifier as a function of the number of features [3]. Since the research in feature selection has been a challenging field, some researchers have doubted about its computational feasibility, such as in the paper [4]. Despite the computationally challenging scenario, the research in this direction continued. As of 1997, several papers on variable and feature selection were published [5,6]; however, few papers dealt with data sets with more than 40 features [1]. Nowadays, it has enjoyed increased attention due to the massive growth of data across many scientific disciplines, such as in genomic analysis [7], text mining [8], to name a few. To deal with these data, feature selection faces some new challenges. Then, it is timely and significant to review the relevant topics to these emerging challenges and give some suggestions to the practitioners. The discussed challenges and topics are listed in Table 1.

Feature selection brings the immediate effects of speeding up a data mining algorithm, improving learning accuracy, and enhancing model comprehensibility. However, finding an optimal feature subset is usually intractable [6] and many problems related to feature selection have been shown to be NP-hard [9]. To efficiently solve this problem, two frameworks are proposed up to now. One is the search-based framework, and the other is the correlation-based framework [2]. For the former, the search strategy and evaluation criterion are two key components. The search strategy is about how to produce a candidate feature subset, and each candidate subset is evaluated and compared with the previous best one according to a certain evaluation criterion. The process of subset generation and evaluation is repeated until a given stopping criterion is satisfied. For the latter, the redundancy and relevance of feature are calculated based on some correlation measure. The entire original feature set can then be divided into four basic disjoint subsets: (1) irrelevant features, (2) redundant feature, (3) weakly relevant but non-redundant features, and (4) strongly relevant features. An optimal feature selection algorithm should select non-redundant and strongly relevant features. When the best subset is selected, generally, it will be validated by prior knowledge or different tests via synthetic and/or real-world data sets. One of the most well-known data repositories is in UCI [10], which contains many kinds of data sets with different sizes of

Table 1 The summary of challenges and topics for current feature selection research

Challenges	Topics
High dimensionality small sample size (HDSSS) data	Topic 1: Stable feature selection
	Topic 2: Sparsity-based feature selection
	Topic 3: Multi-sources feature selection
Big data	Topic 4: Distributed feature selection
	Topic 5: Multi-view feature selection
	Topic 6: Multi-label feature selection
	Topic 7: Online feature selection
Secure feature selection	Topic 8: Privacy-preserving feature selection
	Topic 9: Adversarial feature selection

sample and dimensionality. Feature Selection @ ASU (http://featureselection.asu.edu) also provides many benchmark data sets and source codes for different feature selection algorithms. In addition, some microarray data, such as Leukemia [11], Prostate [12], Lung [13], and Colon [14], are often used to evaluate the performance of feature selection algorithms on the high dimensionality small sample size (HDSSS) problem.

The feature selection algorithms have been surveyed in many papers [1,2,15–17]. However, in recent years, the fast development of machine learning broadens the scopes of feature selection research and applications, and then, it is time to comprehensively study the recent advances of feature selection. We need to point out that [18–20] are three latest review papers on feature selection. The [18] still focuses on the traditional feature selection similar to [2,17] besides the comparison of some classical feature selection algorithms on some real-world data sets. Most of algorithms summarized in [19,20] are under the hybrid (embedded) model. The evaluation function of these algorithms generally consists of a loss function and a L1 or L2,1 regularization term, aiming for sparsity. The features with nonzero weights are selected ones. However, in our survey, besides the topics introduced in [19,20], many recent developments are also reviewed, such as distributed feature selection, stable feature selection, and privacy-preserving feature selection. While [20] summarizes the feature selection algorithms from the data perspective, we review the algorithms from the problem perspective.

The review is organized as follows: some previous related works are introduced according to some frameworks in Sect. 2, and the advanced topics of feature selection are summarized in Sect. 3. Section 4 presents the representative applications of feature selection. The paper ends with future works and conclusion in Sect. 5.

2 Related work

Many feature selection algorithms have been proposed over the past decades. These algorithms are either search-based or correlation-based. We briefly introduce existing algorithms according to these two frameworks.

2.1 Feature selection frameworks

In the following two subsections, we show how feature selection works in each framework and what their strengths and weaknesses are.

2.1.1 Search-based feature selection framework

For the search-based framework, a typical feature selection process consists of three basic steps (shown in Fig. 1), namely subset generation, subset evaluation, and stopping criterion. Subset generation aims to generate a candidate feature subset. Each candidate subset is evaluated and compared with the previous best one according to a certain evaluation criterion. If the newly generated subset is better than the previous one, it will be the latest best subset. The first two steps of search-based feature selection are repeated until a given stopping criterion is satisfied.

Figure 1 indicates that search-based feature selection includes two key factors: the evaluation criterion and the search strategy. According to the evaluation criterion, feature selection algorithms are categorized into filter, wrapper, and hybrid (embedded) models. Feature selection algorithms under the filter model rely on analyzing the general characteristics of data and evaluating features without involving any learning algorithms. Wrapper utilizes a predefined

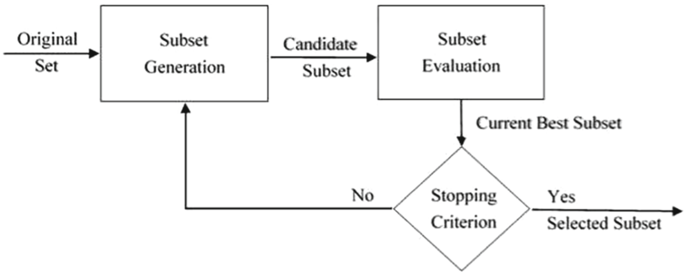


Fig. 1 Search-based feature selection

Table 2 The analysis of search-based feature selection algorithms

Models	Advantages	Disadvantages
Filter	Fast	
	Scalable	Ignores interaction with the classifier
	Independent of the classifier	
Wrapper	Simple	Risk of overfitting
	Interacts with the classifier	Classifier dependent selection
	Models feature dependencies	Computationally intensive
Hybrid (embedded)	Interacts with the classifier	
	Less complexity than Wrapper	Classifier dependent selection
	Models feature dependencies	

learning algorithm instead of an independent measure for subset evaluation. A typical hybrid algorithm makes use of both an independent measure and a learning algorithm to evaluate feature subsets. The analysis of advantages and disadvantages for filter, wrapper, and hybrid model is summarized in Table 2. On the other hand, search strategies are usually categorized into complete, sequential, and random models. Complete search evaluates all feature subsets and guarantees to find the optimal result according to the evaluation criterion. Sequential search likes to add or remove features for the previous subset at a time. Random search starts with a randomly selected subset and injects randomness into the procedure of subsequent search. Some earlier studies have been categorized based on the evaluation criterion and the search strategy in [2].

Nowadays, as the big data with high dimensionality are emerging, the filter model has attracted more attention than ever. Feature selection algorithms under the filter model rely on analyzing the general characteristics of data and evaluating features without involving any learning algorithms; therefore, most of them do not have bias on specific learner models. Moreover, the filter model has straightforward search strategy and feature evaluation criterion, and then, its structure is always simple. The advantages of the simple structure are evident: first, it is easy to design, easy to be understood by other researchers. Second, it is usually very fast [15] and is often appropriate for high-dimensional data.

2.1.2 Correlation-based feature selection framework

Besides the search-based feature selection, another important framework for feature selection is based on the correlation analysis between features and classes. The correlation-based

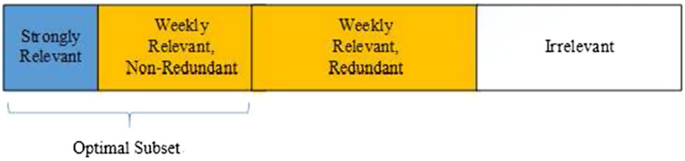


Fig. 2 Optimal subset for correlation-based feature selection

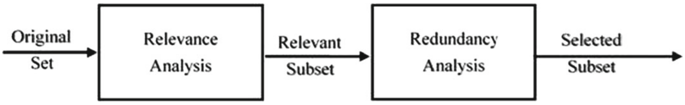


Fig. 3 Correlation-based feature selection

framework considers the feature–feature correlation and feature–class correlation. Generally, the correlation between features is known as feature redundancy, while the feature–class correlation is viewed as feature relevance. Then an entire feature set can be divided into four basic disjoint subsets: (1) irrelevant features, (2) redundant features, (3) weakly relevant but non-redundant features, and (4) strongly relevant features. An optimal feature selection algorithm should select non-redundant and strongly relevant features as shown in Fig. 2. The classical definitions for feature relevance and redundancy are introduced as follows. Let $\mathbf{F} = \{f_1, \ldots, f_d\}$ be a full set of features, C be a full set of class labels, P be the probability distribution, f_j be a feature, and $\mathbf{S}_j = \mathbf{F} - f_j$.

Definition 1 (*Strong relevance*) Given a class C, a feature f_j is strong relevance iff

$$P(C|f_j, \mathbf{S}_j) \neq P(C|\mathbf{S}_j) \tag{1}$$

Definition 2 (*Weak relevance*) Given a class C, a feature f_j is weakly relevant iff

$$\begin{aligned} &P(C|f_j, \mathbf{S}_j) = P(C|\mathbf{S}_j), \quad \text{and} \\ &\exists \mathbf{S}'_j \subset \mathbf{S}_j, \quad \text{such} \quad \text{that} \quad P(C|f_j, \mathbf{S}'_j) \neq P(C|\mathbf{S}'_j) \end{aligned} \tag{2}$$

Definition 3 (*Irrelevance*) Given a class C, a feature f_j is irrelevant iff

$$\forall \mathbf{S}'_j \subset \mathbf{S}_j, \quad P(C|f_j, \mathbf{S}'_j) = P(C|\mathbf{S}'_j) \tag{3}$$

Definition 4 (*Redundancy*) Let **f** be the current selected feature subset, a feature f_j is redundant iff it is weakly relevant and has a Markov blanket M_j within **f**. And M_j is said to be a Markov blanket for f_j iff

$$P(\mathbf{F} - M_j - \{f_j\}, C|f_j, M_j) = P(\mathbf{F} - M_j - \{f_j\}, C|M_j) \tag{4}$$

The correlation-based feature selection framework is shown in Fig. 3, which consists of two steps: relevance analysis determines the subset of relevant features, and redundancy analysis determines and eliminates the redundant features from relevant ones to produce the final subset. This framework has advantages over the search-based framework as it circumvents subset search and allows for an efficient and effective way in finding an approximate optimal subset. The most well-known feature selection algorithms under this framework are mRMR [21], Mitra's [22], CFS [23], and FCBF [24,25].

In Sect. 2, we briefly summarize the earlier studies on feature selection. For more details about the basic knowledge of feature selection and the algorithms mentioned above, please refer to [1,2,15–17]. In the following section, we will focus on the recent advances on feature selection along with the newly emerged challenges in data processing.

3 Advanced topics for feature selection

The massive amounts of high-dimensional data bring about both opportunities and challenges to feature selection. Valid computational paradigms for new challenges are becoming increasingly important. Then along with the paradigms, many feature selection topics are emerging, such as feature selection for high dimensionality small sample size (HDSSS) data, feature selection for big data mining, feature selection for multi-label learning, feature selection with privacy preservation, and feature selection for streaming data mining. So we like to survey these new ongoing topics and corresponding solutions in this section.

3.1 Feature selection for high dimensionality small sample size data

One challenging scenario in many feature selection applications is the HDSSS, where the dimensionality of data is extremely high, while the sample size is very small. The related topics of feature selection for the HDSSS data include stable feature selection, sparsity representation and multi-sources feature selection.

Topic 1: Stable feature selection

Problem Various feature selection algorithms have been developed with a focus on improving classification accuracy while reducing the dimensionality. Furthermore, another important issue about the stability of feature selection recently attracted much attention. The stability means the insensitivity of the result of a feature selection algorithm to variations of the training set [26–28]. When feature selection is adopted to identify the critical markers to explain some phenomena, the stability is very important. For instance, in microarray analysis, biologists are interested in finding a small number of features (genes or proteins) that explain the mechanisms driving different behaviors of microarray samples. A feature selection algorithm is often used to choose a subset of genes or proteins. However, the selection results will be largely different under variations to the training data, although most of results are similar to each other in terms of the classification performance [28]. Such instability dampens the confidence of domain experts in experimentally validating the selected features.

In consideration of the importance of stability in applications, several stable feature selection algorithms have been proposed, such as ensemble methods [26,29–31], samples weighting [27,32], and feature grouping [28,33], to name a few. A comprehensive survey on stable feature selection can be found in [34]. We like to introduce these solutions in detail as follows:

Solution 1 Ensemble methods. Ensemble feature selection techniques use an idea similar to ensemble learning for classification: In the first step, a number of different feature selectors are produced, and in a final phase, the outputs of these separate selectors are aggregated and returned as the final (ensemble) result. Variation in the feature selectors can be achieved by various methods: choosing different feature selection techniques, instance-level perturbation (e.g., by removing or adding samples), feature-level perturbation (e.g., by adding noise to features), stochasticity in the feature selector, Bayesian model averaging, or combinations of these techniques. Aggregating different feature selection results can be done by weighted voting, e.g., in the case of deriving a consensus feature ranking, or by counting the most frequently selected features in the case of deriving a consensus feature subset [26].

In this paper, we present the ensemble feature weighting as an example and the framework is shown in Fig. 4. The Bootstrap-based strategy is used to train base feature selectors on m different bootstrap subsets of the original training set D. Ensemble feature weighting result

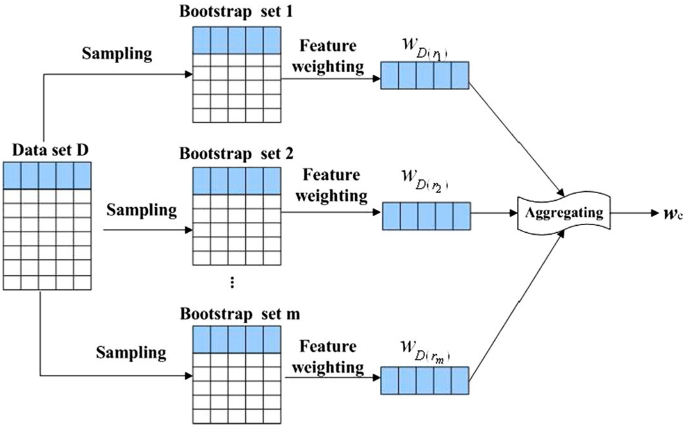


Fig. 4 Ensemble feature weighting

is achieved by averaging the obtained outputs $\mathbf{w}_{D(\mathbf{r}_t)}$ $(t = 1, \ldots, m)$ from the base feature selectors. The theoretical analysis and experimental results about the stability of ensemble feature weighting are presented in our work [35,36].

Solution 2 Sample weighting. The main idea of this solution is to assign different weights to each sample in the training set, and the weight depends on sample's influence to feature relevance estimation. Then it provides a weighted training set to train feature selection method. The key point in this solution is how to determine the sample influence on feature relevance. The influence is always measured by the samples' view or local profile of feature relevance. The basic idea behind the local profile is that the central mass region containing most of the instances is clearly more important in deciding the aggregate feature weight than the outlying region with a couple of outliers. Following the ideas of importance sampling, instances with higher outlying degrees from central mass region should be assigned lower instance weights, which leads to reduce the variance of feature weighting under training data variations. If a sample shows a noticeably distinct local profile from other samples, its absence or presence in the training data will substantially affect the feature selection result. In order to improve feature selection stability, samples with remote local profiles need to be weighted differently from other samples. In [27,32], the local profile of feature relevance for a given sample is measured based on hypothesis margin, and then, a margin-based sample weighting algorithm is proposed to assign a weight to each sample according to the remote degree of its local profile. In [27,32], for a sample $\mathbf{x}_i$, its hypothesis margin is always determined by the distance among $\mathbf{x}_i$, its nearest neighbor with same class label and its nearest neighbor with different class label. An intuitive interpretation of hypothesis margin is a measure of the proportion of the features in $\mathbf{x}_i$ that can be corrupted by noise (or how much $\mathbf{x}_i$ can "move" in the feature space) before $\mathbf{x}_i$ is being misclassified [37]. Of course, the local profile of a sample can be calculated using other measures, and it is an ongoing research.

Solution 3 Feature grouping. It is motivated by a key observation that intrinsic feature groups (or groups of correlated features) commonly exist in high-dimensional data, and such groups are resistant to the variations of training samples. Then, it is natural to select stable features through determining the features groups. Usually, we can approximate intrinsic feature groups by a set of consensus feature groups and perform feature selection in the transformed feature space described by consensus feature group. Generally, the (ensemble) clustering algorithm can be adopted to identify the consensus feature groups. For example, the DRAGS is proposed in [33] to identify dense feature groups based on kernel density estimation, and treat features in each dense group as a coherent entity for feature selection. Another feature grouping method is introduced in [28] based on ensemble idea, which has two essential steps in identifying consensus feature groups:

Step 1. To create an ensemble of feature grouping results;
Step 2. To aggregate the ensemble into a single set of consensus feature groups.

In Step 1, the Dense Group Finder (DGF) in DRAGS algorithm is considered as the base algorithm for identifying feature groups and is applied on a number of bootstrapped training sets from a given training set. The result of this step is an ensemble of feature groupings. In Step 2, a given ensemble of feature groupings is aggregated into a final set of consensus groups. The aggregation strategy is resorted to instance-based ensemble clustering or cluster-based ensemble clustering [38].

Discussion For the solutions introduced above, Solutions 1 and 2 utilizes the label information, and then, they are supervised stable feature selection solutions. Solution 3 does not rely on the label information, and it belongs to unsupervised learning. Then, one future work for stable feature selection is to combine supervised methods with unsupervised ones to obtain semi-supervised stable feature selection methods. We also observe that ensemble is a widely adopted idea to improve the stability of feature selection, such as in Solutions 1 and 3. However, the ensemble strategies used in Solutions 1 and 3 are borrowed from classification or clustering ensemble. Then, specific ensemble strategy for stable feature selection should be proposed in future. Lastly, for all the aforementioned stable feature selection solutions, stability validation mainly depends on the experiment, and the insight of feature selection stability is not explored completely. Although we have done some primary theoretical works on the stable feature weighting in [36], the stability of feature ranking and feature subset selection is not investigated. As a result, the theoretical analysis for stable feature selection still needs more attention.

Topic 2: Sparsity-based feature selection

Problem For the HDSSS data, the dimensionality is extremely high, while the sample size is very small. Sparsity-based feature selection is an efficient tool to select features from HDSSS data. The basic idea of sparsity-based feature selection is to impose a sparse penalty to select discriminative features. The L1-norm (namely Lasso, least absolution shrinkage and selection operator) [39] is effectively implemented to make the learning model both sparse and interpretable. Feature selection with L1-norm has been fully analyzed in [40]. However, Lasso tends to select only one of the pairwise correlated features and cannot induce the group effect. Then, the Lasso should be largely improved.

Solution 1 In order to remedy the deficiency of Lasso and due to the importance of structural information, such as group, recently a general definition of the structured sparsity-inducing norm is proposed to incorporate the prior knowledge or structural constraints to find the suitable linear features [41]. Group Lasso [42] and elastic net [43], and even the tree-guided group Lasso [44] are under the setting of structured sparsity-inducing norm. A very popular and successful approach to learn linear classifiers $\mathbf{w} = \{w_1, \ldots, w_d\}$ with structured features is to minimize a regularized empirical loss. For a give data set $D = \{\mathbf{X}, \mathbf{Y}\} = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$, we choose a predictor $\mathbf{w}$ by minimizing the following empirical loss with regularized term,

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} L(\mathbf{w}^T \mathbf{x}_i, y_i) + \gamma \mathcal{R}(\mathbf{w}, U) \tag{5}$$

where $L(\cdot)$ is the loss function. Popular choices of $L(\cdot)$ are least squares, hinge and logistic loss. $\mathcal{R}(\cdot)$ is a regularization term, and γ is the regularization parameter controlling the trade-off between the $L(\cdot)$ and the regularization. U denotes the structure of features, which includes group structure, tree structure, and graph structure. The value of elements in the learned classifier $\mathbf{w}$ can be equal to zero. Since each element in $\mathbf{w}$ corresponds to one feature, such as w_j corresponds to feature f_j, $(j = 1, \ldots, d)$, feature selection then chooses the features correspond to elements with nonzero value in $\mathbf{w}$. The detailed summary for group structure, tree structure, and graph structure information used in the sparsity-based feature selection is presented in [19].

Solution 2 Besides the sparsity-based feature selection embedded structure information, another notable advance is the implementation of safe feature screening before sparsity-based feature selection. The safe feature screening intrinsically belongs to feature selection. For large-scale problems, solving the L1 regularized sparsity with higher accuracy remains challenging. One promising solution is first to discard ("screening") the "inactive" features. This would result in a reduced feature matrix for sparsity-based feature selection and save the computational cost and memory size. A fast and effective sparse logistic regression screening rule (Slores) to identify the "inactive" features is proposed in [45]. The proposed screening rule detects "inactive" features by estimating an upper bound of the inner product between each feature vector and the "dual optimal solution" of the L1 regularized logistic regression. The safe screening rule has been applied into multi-task feature learning [46].

Discussion The sparsity-based feature selection has gained much attention in machine learning and statistics. For more details about the sparsity-based feature selection, please refer to the latest survey [19,20]. The sparsity strategy is also adopted in many other feature selection topics, such as online feature selection described below.

Topic 3: Multi-source feature selection

Problem The small sample size in the HDSSS problem has negative influence on the reliability of statistical analysis. An alternative way to address this issue is to utilize additional information sources to enhance our understanding of the data in hand, which leads to multi-source feature selection. How to extract and represent useful knowledge for feature selection from different sources and then obtain the uniform result is one key problem in multi-source feature selection. As summarized in [15,47], the knowledge used in feature selection usually can be categorized into two kinds: the knowledge about features and the knowledge about samples. The former usually contains information about the properties of features or their relationships, and the latter usually contains sample categories or their similarity.

Solution 1 One framework for multi-source feature selection introduced in [15] is shown in Fig. 5 where the heterogeneous knowledge is commonly represented by similarity among samples via conversion operation, and the spectral feature selection algorithm is applied. The framework can be summarized as follows:

(1) Knowledge conversion. The conversion operator is adopted to extract a local specification of sample similarity matrix for each knowledge source;

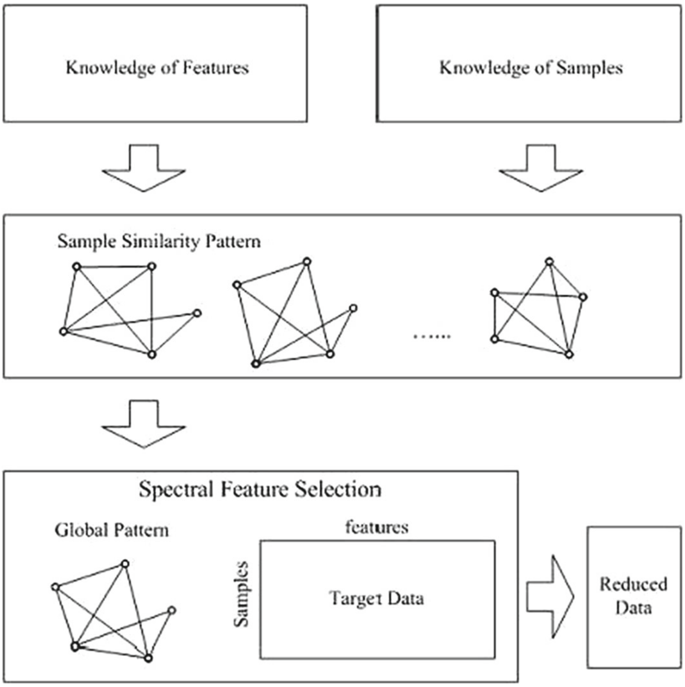


Fig. 5 The framework for multi-sources spectral feature selection

(2) Knowledge integration. The multiple local sample similarity matrices are linearly combined to obtain a global similarity matrix;
(3) Feature selection. A spectral feature selection algorithm is performed on the obtained global similarity matrix.

Note that the different knowledge sources need different conversion operations. For example, if the similarity among features is given, then the feature covariance can be constructed and used in calculating the pairwise sample similarity via Mahalanobis distance [15].

Solution 2 The above framework depends on combining local sample similarity matrix and a spectral feature selection algorithm has to be used, which is not flexible for the choice of other feature selection algorithms in handling small sample data. Then, another general framework—KOFS—is presented in [47] to address this limitation through combining the local feature selection results from different knowledge sources. KOFS consists of three components described below and is shown in Fig. 6.

(1) Knowledge conversion. Transform different types of human or external knowledge to certain types internal knowledge that can be used by feature selection algorithms.
(2) Feature ranking. Rank the features based on the internal knowledge and some given criterion. And for multiple knowledge, we can obtain several feature ranking results.
(3) Rank aggregation. Combine the multiple feature ranking results to generate the final ranking.

Discussion Multi-source feature selection aims to utilize various knowledge sources to advance feature selection research on the HDSSS problems. The key problem for multi-

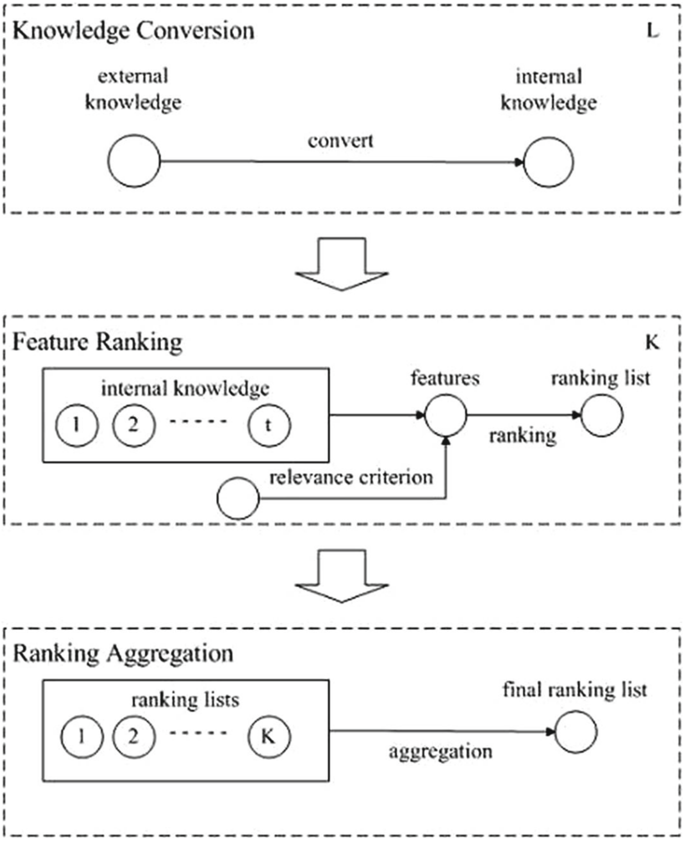


Fig. 6 The framework of KOFS

source feature selection is knowledge conversion. We should convert the heterogeneous knowledge from different sources into samples or features knowledge that can be utilized by a feature selection method. Then, the knowledge conversion depends on the knowledge source and the feature selection algorithm. On the other hand, the usefulness of knowledge sources for feature selection and the combination of feature selection result from different sources should be discussed.

3.2 Feature selection for big data

With the rapid development of the Internet, big data of large volume and ultrahigh dimensionality has emerged in various machine learning applications. For instance, Weinberger et al. [48] have studied a collaborative spam email filtering task with 16 trillion (10^{13}) unique features. With the tremendous growth of data set sizes, the scalability of most current feature selection algorithms may be jeopardized. To improve the scalability of a feature selection algorithm, the distributed computing strategy is always adopted. In addition, in big data era, we can describe the data information based on multiple views, which are corresponding to different knowledge sources. Then, multi-view feature selection is another topic relevant with big data.

Topic 4: Distributed feature selection

Given a large-scale data set containing a huge number of samples and features, the scalability of a feature selection algorithm becomes extremely important. However, most existing feature selection algorithms are proposed to deal with the data sets whose sizes are under several gigabytes. In order to improve the scalability of existing feature selection methods for large-scale data sets, the feature selection is always performed in distributed manner. It has been shown in [49] that any operation fitting the Statistical Query model can be computed in parallel based on data partitioning. Studies also showed that when the data size is large enough, parallelization based on data partition can result in linear speedup as computing resources increase [49].

Problem 1 When the data are located at a central database, how to implement the distributed feature selection.

Solution For a central data repository, efficient distributed programming models and protocols, such as MPI [50] and Google's MapReduce [51], are resorted. These models help implement the distributed feature selection on high-performance computer grids or clusters. In general, feature selection can be parallelized in four steps as follows: (1) decompose the feature selection process into summation forms over training samples, (2) divide data and store data partitions on nodes of the cluster, (3) compute local feature selection results in parallel on nodes of the cluster, and (4) calculate the final feature selection result by integrating the local results.

For example, the spectral feature selection is parallelized on MPI as described in [52]. In addition, a novel large-scale feature selection algorithm is proposed in [53]. The algorithm chooses features by evaluating their abilities to explain data variances. The algorithm can read data in distributed form and perform parallel feature selection in both symmetric multiprocessing mode and massively parallel processing mode. The algorithm has been developed as a SAS High-Performance Analytics procedure.

Problem 2 On the other hand, data are not located at a central repository, and rather distributed across a large number of nodes in a network. The next generation of Peer-to-Peer (P2P) networks provides an example. The existing feature selection algorithms cannot be directly executed under these scenarios. Therefore, analysis of data in such networks will require the development of another type of distributed feature selection algorithm capable of working in such large-scale distributed environments.

Solution In [54], a distributed feature selection algorithm on P2P network is introduced. The proposed P2P feature selection algorithm (PAFS) incorporates three popular feature selection criteria: misclassification gain, gini index, and entropy measurement. As a first step, these measurements are evaluated in a P2P network without any data centralization. Then, the algorithm works based on local interactions among participating peers. The algorithm is provably correct in the sense that it converges to the correct result compared to centralization. And the proposed algorithm has low communication overhead.

Discussion When data are distributed across a large number of nodes in a network, a distributed feature selection algorithm has to enable an information exchange and fusion mechanism to ensure that all distributed sites (or information sources) can work together to achieve a global optimization goal [55]. Local feature selection and correlations are the key steps to ensure that the results discovered from multiple information sources can be consolidated to meet the global objective. For the current distributed feature selection, the data in each node or peer have the same set of features. However, more studies are needed in the distributed feature selection when the data in different nodes may have different feature representations.

Topic 5: Multi-view feature selection

Problem In the era of big data, we can easily access information from different sources. For example, in medical scenario, measurements from a series of medical examinations are documented for each subject, including clinical, imaging, immunologic, serologic, and cognitive measures. These measurements are obtained from multiple sources, and each measurement is a view of subject. It is desirable to combine all these measurements for multi-view learning in an effective way. However, some measurements are irrelevant or noisy, even conflicting, which is unfavorable for multi-view learning. To address this issue, feature selection should be incorporated into the process of multi-view learning.

Solutions Three strategies of multi-view feature selection are summarized in [56]: (1) concatenates features from all views in the input space, and then a traditional feature selection method is adopted; (2) converts multiple views into a tensor and directly performs feature selection in the tensor product space; (3) efficiently conducts feature selection in the input space while effectively leveraging relationships between the original data and its reconstruction in the tensor product space. Methods (1) and (2) ignore the intrinsic properties of raw multi-view features and hidden relationships between the original data and its reconstruction. The problem of feature selection in the tensor product space is formulated as an integer quadratic programming problem in [57]. However, this method is limited to the interaction between two views and is hard to be extended to many views, since it directly selects features in the tensor product space resulting in the curse of dimensionality. Tang et al. [58] study multi-view feature selection in the unsupervised setting. Taking into account the latent interactions among views in [56], tensor product is used to organize multi-view features, and [56] studies the problem of multi-view feature selection based on SVM-RFE [59] and tensor techniques.

Discussion In the era of big data, multi-view learning is an useful paradigm to deal with heterogeneous data. The multi-view feature selection is different from the multi-source feature selection above. If multi-source feature selection is adopted, feature selection is always conducted independently for each resource and the results from multi-source are combined to obtain the final result. This feature selection paradigm always assumes each source is sufficient to learn the target concept and ignore the relationship between sources. However, in multi-view learning, individual view can often provide complementary information to each other and does not assume each view is sufficient to learn target concept, which lead to the improved performance in real-world applications. Moreover, the relationship between views is considered in multi-view learning, and tensor product space is always used to represent this relationship. In summary, multi-source feature selection is a special case for multi-view feature selection. On the other hand, the multi-view learning is also relevant with the cross-domain learning, i.e., to transfer the already learned knowledge from a source domain to a target domain. Then, multi-view feature selection can be applied into transfer learning by taking advantage of the typically available multiple views of the data in domains [60].

Topic 6: Multi-label feature selection

Problem In traditional feature selection, the training sample always has a unique label. However, in many real applications, each instance can be associated with more than one class label. Then, feature selection for multi-label data also attracts much attention.

Solution 1 The simplest way to implement feature selection on multi-label data set is based on transformation, i.e., change the multi-label data set into a single-label one and a traditional feature selection method is used. There are a lot of strategies to transform a multi-label data set into a single-label one, such as,

(1) Simple transformation. Its main idea is to convert a multi-label data set into a single-label one by some label selection operator, such as selecting the most frequent label in the data set (select-max), the least frequent label (select-min), a random label (select-random), simply discard every multi-label example (select-ignore).
(2) Copy transformation. This transformation copies each multi-label instance B times, where B is the number of labels associated with that instance. Each copied instance is then assigned one distinct single label from the original B-label set. A slight improvement of the copy transformation is the weighting copy, which assigns a weight $\frac{1}{B}$ to each copied instance.
(3) Label powerset transformation. Label powerset (LP) considers each subset of labels that exists in the multi-label training data set as one label.
(4) Binary relevance transformation. Binary relevance (BR) creates a binary training set for each label in the original data set. Besides the classic transformations above, a new transformation based on the entropy measure is presented in [61], where an instance is assigned to a label according to a certain probability. Then, after the data transformation step, many single-label feature selection techniques can be used, for example, information gain [62], Chi-square statistic [63], and Orthogonal Centroid Feature Selection (OCFS) [64].

Solution 2 The multi-label feature selection methods based on transformation usually suffer from the deficiency that the correlation among the class labels is not taken into account. Furthermore, when there are lots of class labels, the obtained single-label data sets may be too large. Then, each imbalanced single-label data set will face the problems of sparse training samples and imbalance class distribution. So it is desirable to propose feature selection algorithms to deal directly with multi-label data. Most current methods are adaptations of well-known single-label feature selection algorithms. For example, in [65], the well-known algorithm Fast Correlation-Based Filter (FCBF) [24,25] is extended to handle multi-label data through graphical model to represent the relationships among labels and features. Another example is presented in [66], where a multi-label feature selection approach gMLC is designed for graph classification. It is based on an evaluation criterion to estimate the dependence between subgraph features and multiple labels of graphs. Then, a branch-and-bound algorithm is proposed to efficiently search for optimal subgraph features by judiciously pruning the subgraph search space using multiple labels. Moreover, a multi-label feature selection method is also described in [67], which is based on the minimization of correlation regularized loss of label ranking [68]. In our recent work [69], we adopted the graph model to capture the label correlation, and propose a multi-label feature selection algorithm based on graph and large margin theory.

Discussion Most of current multi-label feature selection methods are the extension of traditional single-label feature selection approaches. A future research is to design multi-label feature selection that it can directly deal with multi-label data set without any transformation and sufficiently consider the relationship between labels.

Topic 7: Online feature selection

Problem 1 All feature selection methods introduced above assume that all features are known in advance. Now another interesting scenario should be considered where candidate features are sequentially presented to the classifier. In this scenario, the candidate features are generated dynamically and the size of features is unknown. This kind of feature selection for streaming features is also called online feature selection. Online feature selection is significant in many applications. For example, the famous microblogging Web site Twitter produces more than 250 millions tweets per day and many new words (features) are generated such as abbreviations. When we like to select features for tweets, it is impossible to wait until all features have been generated; thus, it could be more preferable to online feature selection.

Solution Up to now, some online feature selection methods have been proposed in [19,70–74]. In general, an online feature selection algorithm will perform the following steps [19],

Step 1: Generating a new feature,
Step 2: Determining whether the newly generated feature should be added to currently selected feature subset,
Step 3: Determining whether some features should be removed from the currently selected feature subset when the new feature is added,
Step 4: Repeat Steps 1–3.

Note that different algorithms may have different implementations for Steps 2 and 3. Step 3 is optional and some online feature selection algorithms only implement Step 2. The correlation model introduced in Sect. 2 is always used in online feature selection to calculate the new feature's relevance and redundancy, which will determine the new feature's inclusion, and the selected features' deletion.

The general online feature selection algorithms often assume features arrive one by one at a time; however, in some practical applications such as image analysis and email spam filtering, features may arrive by groups. Then, the online feature selection should consider the group information. A novel selection approach is proposed in [75] to solve the online group feature selection problem. The proposed approach consists of two stages: online intra-group selection and online inter-group selection. In the intra-group selection, spectral analysis is used to select discriminative features in each group when it arrives. In the inter-group selection, Lasso is adopted to select a globally optimal subset of features. This two-stage procedure continues until there are no more features arrive or some predefined stopping conditions are met.

Problem 2 On the other hand, we also encounter another scenario that training instances arrive sequentially, while all features are available before the selection process. Online feature selection in this scenario aims to select a small and fixed number of features in an online learning fashion.

Solution Take this problem into account, an approach for this online feature selection is proposed in [76]. When the learner received a training instance at each time, the classifier will immediately make a prediction of the instance. When the training instance is misclassified, the classifier is first updated by online gradient descent and then projected to a L_2 ball to ensure that the norm of the classifier is bounded. If the resulting classifier has more than K nonzero elements in weight vector (K is the fixed number of selected features), the algorithm will take a truncate technique, simply keeping the K elements in classifier with the largest absolute weights.

Problem 3 A new problem for online feature selection is to learn from doubly streaming data where both data volume and feature space increase over time. This problem is always referred as mining trapezoidal data streams, to which existing online learning, online feature selection and streaming feature selection algorithms are inapplicable.

Solution A new Sparse Trapezoidal Streaming Data mining algorithm (STSD) that combining online learning and online feature selection to enable learning trapezoidal data streams with infinite training instances and features is introduced in [77]. Specifically, when new training instances carrying new features arrive, the classifier updates the existing features by following the passive-aggressive update rule used in online learning and updates the new features with the structural risk minimization principle. Feature sparsity is also introduced using the projected truncation techniques. The first challenge is to update the classifier with an augmenting feature space. The classifier update strategy is able to learn from new features. The second challenge is to build a feature selection method to achieve a sparse but efficient model, i.e., sparsity strategy.

The update strategy: On each round, when received a training instance, the classifier will immediately predict the instance and compute an instantaneous loss of the prediction using some loss function, such as hinge-loss. In order to make the classifier to learn from new features, the weight vector of classifier is divided into two parts, one part represents a projection of the original feature space and the other denotes new features that are in current round while not in last round. If the prediction is correct, there is not update of current weight vector. Otherwise, the constrained optimization problem is obtained as the solution to the update strategy. On the one hand, the constrained problem forces the classifier to predict correctly. On the other hand, the constrained problem forces the projection of feature space close to the weight vector obtained in last round with the aims to inherit information and let the weights of new features be small to minimize structural risk and avoid overfitting.

The sparsity strategy: the algorithm takes truncate technique by introducing a parameter to control the proportion of features. Besides, the algorithm introduces a projection step by projecting the weight vector to an L_1 ball because one single truncation step does not work well.

Discussion In the big data era, the streaming feature and sample will be encountered in many applications, such as financial analysis, online trading, and medical testing. Static feature selection methods cannot adapt to the characteristics of dynamic data streams, such as continuity, variability, rapidity, and infinity, and can easily lead to the loss of useful information. Therefore, effective theoretical and technical frameworks are needed to support online feature selection. The online feature selection simultaneously considering the streaming feature and streaming sample is an active research topic.

3.3 Secure feature selection

Currently, many collected data for pattern recognition and data mining are highly sensitive, such as medical details, census records, bank statements, and inventory records. These data reveal the intimate details in our daily life. So the analysis of private information often raises concerns regarding the privacy rights of individuals and organizations. However, most feature selection methods do not address the information security issues. Then, secure feature selection should be put more attention. Among the different secure concerns, the privacy-preserving feature selection and adversarial feature selection are our focus.

Topic 8: Privacy-preserving feature selection

Problem 1 The goal of privacy-preserving feature selection is to find a feature subset that a classifier will minimize the classification error in the selected feature space, and the sum of privacy degree of features in the selected subset will not beyond the predefined threshold [78]. There exist two key issues in this kind of privacy-preserving feature selection: one is the determination of privacy degree for each feature and the other is the optimization of evaluation criterion with privacy constraint.

Solution A privacy-preserving feature selection method was proposed and applied into face detection in [78]. For the determination of privacy degree for each feature, the principal component analysis (PCA) spectrum is used to measure the amount of privacy information in [78]. And specifically, the PCA space of all the face images in the database is computed and maps all the data to that space without reducing dimensionality. Then, the privacy degree of all features is set to the eigenvalues associated with each dimension in the PCA space. For the optimization of evaluation criterion with privacy constraint, in the [78], a variant of the gentleBoost algorithm [79] is adopted to find a greedy solution to constrained objective function for privacy-preserving feature selection. Specifically, they use gentleBoost with "stumps" as the weak classifiers where each "stump" works on only one feature. In each iteration, they can use features that were already selected or those that adding them will not increase the total degree of selected features beyond the privacy threshold.

Discussion One key problem in privacy-preserving feature selection above is the determination of feature privacy degree, which is often determined by domain knowledge or experts. In this aspect, it is similar to cost-sensitive feature selection where the cost of each feature also should be given by domain knowledge or experts [80]. On the other hand, the evaluation criterion of privacy-preserving feature selection described above is the classification error. Of course, other criteria introduced in Sect. 2 also can be used.

Problem 2 The above privacy-preserving feature selection focuses on the feature privacy and produces the optimal feature subset with the total privacy degree less than a threshold. There still exists another kind of privacy-preserving feature selection that focuses on the sample privacy.

Solution The widely used privacy model for this kind of privacy preservation is differential privacy as in Definition 5 [81].

Definition 5 A randomized mechanism A provides ε-*differential privacy*, if, for all data sets D and D' which differ by at most one element, and for all output subsets $S_u \subseteq Range(A)$,

$$Pr\,[A(D) \in S_u] \leq \exp(\varepsilon) \times Pr\left[A(D') \in S_u\right]. \tag{6}$$

Then, the algorithm A is called satisfying differential privacy. The probability Pr is taken over the coin tosses of A, and $Range(A)$ denotes the output range of A. The privacy parameter ε measures the disclosure. The differential privacy means that an adversary, who knows all but one entry of the data set, cannot gain much additional information about this entry by observing the output of the algorithm.

Based on the privacy model above, two privacy-preserving feature selection algorithms are proposed in our recent work [82,83]. These algorithms are based on output perturbation and objective perturbation, respectively. The output perturbation means adding noises to the feature selection result, and the noise density depends on the sensitivity of feature selection algorithm. In the paper [82], the sensitivity of local learning-based feature selection with logistic loss function is analyzed and the proof for meeting ε differential privacy is also given. Objective perturbation aims to enforce the perturbation on the evaluation function of feature selection, and the corresponding privacy-preserving local learning-based feature selection with the associated differential privacy proof is also presented in [83].

Discussion Certainly, we can enforce privacy-preserving constraints to other traditional feature selection algorithms besides local learning-based feature selection [84]. Current works focus on the output of feature selection is feature weighting, how to perturb other outputs of feature selection, such as feature ranking and feature subset, is an interesting work.

Topic 9: Adversarial feature selection

Pattern recognition and machine learning techniques have been increasingly applied into information security area, such as spam, intrusion, and malware detection. Then, we need to analyze the security of machine learning itself [85]. Correspondingly, the adversarial machine learning is always mentioned. Adversarial machine learning is the design of machine learning algorithms that can resist some sophisticated attacks, and the study of the capabilities and limitations of attackers [86]. These sophisticated attacks include avoiding detection of attacks, causing benign input to be classified as attack input, launching focused or targeted attacks, or searching a classifier to find blind-spots in the algorithm.

Problem The previous works focus on adversarial classification and clustering [87]; only few authors have considered the adversarial feature selection. In fact, for adversarial tasks, feature selection can open the door for the adversary to evade the classification system and misclassify the adversary sample as normal one [88].

Solution Current adversarial feature selection works are concerned with two issues: attack and defense. To explore the vulnerability of some classical feature selection algorithms under attacks [89], sheds light on the issue whether feature selection may be beneficial or even counterproductive when training data are poisoned by intelligent attackers. It also provides a framework to investigate the robustness of popular feature selection methods, including LASSO, ridge regression and the elastic net, to carefully crafted attacks. Another issue discussed in [90] is to analyze the impact of feature selection result on classifier security against the evasion attack where the attackers goal is to manipulate malicious data at test time to evade detection. The basic idea of evaluation criterion for this kind of adversarial feature selection is to select a feature subset that not only maximizes the generalization capability of the classifier, but also its security against evasion attacks. Then, there are two terms in the criterion: generalization capability and security. The generalization capability of a classifier on a feature subset can be estimated using different performance measures as described in Section 2. As for the security term, the robustness against attack is always exploited, i.e., the average minimum number of modifications to a malicious sample to evade detection.

Discussion Many attacks have been introduced in machine learning systems [85]. Currently, only part of particular attacks are considered in adversarial feature selection. So it is urgent to analyze the vulnerability of many classical feature selection under different attacks, and design robust feature selection algorithm against these attacks.

4 Representative applications

Feature selection is a very important preprocessing or knowledge discovery tool for data analysis, and it has been applied into many domains. Then after the review of some recent advances in feature selection, we now introduce some representative applications of feature selection, such as Bioinformatics, social media, and multimedia.

4.1 Bioinformatics applications

During the last decade, the motivation for applying feature selection techniques in bioinformatics has shifted from being an illustrative example to becoming a real prerequisite for model building. Feature selection has been applied into sequence analysis, microarray analysis and mass spectra analysis, single-nucleotide polymorphism analysis, and text and medical literature mining. In particular, the high-dimensional nature of data in bioinformatics has given rise to a wealth of feature selection techniques being presented in the field. A comprehensive survey of earlier work can be found in [91,92].

The feature selection in bioinformatics is usually to solve the HDSSS problem. So the feature selection methods related to the HDSSS problem as introduced in Sect. 3.1 can be applied into Bioinformatics. For example, in [29], an ensemble feature selection method is adopted to identify biomarker for cancer diagnosis. The authors focus on the analysis of ensemble feature selection techniques using linear SVMs and Recursive Feature Elimination (RFE) as the feature selection mechanism. In the first step, by drawing (with replacement) different bootstrap sub-samples of the training data, RFE is applied to each of these bootstrap sub-samples, and thus obtain a diverse set of feature rankings. In the final step, in order to aggregate the different rankings into a final result, two aggregation schemes are used. One is Complete Linear Aggregation (CLA), which uses the complete ranking of all features to create the ensemble result. The ensemble ranking is then obtained by just summing the ranks over all bootstrap sub-samples. Another is Complete Weighted linear Aggregation (CWA), which is a variation on the CLA and assign the weight to each bootstrap ranking. The weight is based on the AUC obtained by a linear SVM trained on the bootstrap samples and evaluated on the out-of-bag samples. Besides the ensemble feature selection, other stable feature selection methods, such as sample weighting [32], are also used in bioinformatics.

Another line for feature selection application in bioinformatics is the use of sparsity regularization related to Topic 2. The most well-known work is introduced in [93], which employs joint $L_{2,1}$-norm minimization on both loss function and regularization. A $L_{2,1}$-norm regularization is performed to select features across all data points with joint sparsity, i.e., each feature (gene expression or mass-to-charge value in mass spectrometry in bioinformatics) either has small scores for all data points or has large scores over all data points. Extensive experiments have been performed on six bioinformatics data sets and the experimental results shown its performance outperforms five other commonly used feature selection methods in statistical learning and bioinformatics. We also like to point out that the multi-source feature selection algorithms in [15,47] are originally designed for bioinformatics application.

4.2 Social media applications

The social media services, such as Facebook and Twitter, are very popular in recent years. And these media supply a very convenient manner for people's communication. The big social media data with high dimensionality pose new challenges to data mining tasks such as classification and clustering. One approach to reduce the dimensionality of social media data is still feature selection [94–96].

To apply the feature selection into social media, the domain knowledge should be embedded. One of the domain knowledges has been considered is the link information between user and user or between user and posts (e.g., tweets, blogs, or images) in the context of social media. To utilize this knowledge, two fundamental problems should be studied in the feature selection, which are (1) the relation extraction from linked data including labeled data and unlabeled data and (2) the mathematical representation for these relations. To address the issues in supervised case [94], introduced four types of relations, such as (a) a user can have multiple posts, (b) two users follow a third user, (c) two users are followed by a third user, and (d) a user follows another user. And these relations are represented by four corresponding hypotheses, such as (a) CoPost Hypothesis: This hypothesis assumes that posts by the same user are of similar topics, (b) CoFollowing Hypothesis: This hypothesis suggests that if two users follow the same user, the topics of their posts are likely similar, (c) CoFollowed Hypothesis: It says that if two users are followed by the same user, their posts are similar in topics, and (d) Following Hypothesis: The hypothesis assumes that one user follows another because they have same interests. Thus, their posts are more likely similar in terms of topics. To integrate these hypotheses into some common feature selection evaluation criterion, regularization terms, such as $L_{2,1}$-norm, embedded with the hypotheses are added.

Another line of feature selection in social media is unsupervised one in [95,97]. For the unsupervised feature selection in social media, the relation extracted from unlabeled data consists of (1) the social dimension for the linked data and (2) the constraint of attribute value. The social dimension is extracted based on Modularity Maximization [98]. Instances in the same social dimension are similar and instances from different social dimensions are dissimilar. The constraint of attribute value is extracted according to spectral analysis [99]. By introducing the concept of pseudo-class labels, a unsupervised feature selection framework—LUFS—for social media is proposed in [95,97]. The feature selection is completed in this framework through optimizing these two relations extracted from unlabeled data. On the other hand, to address some issues, such as the dynamic feature generation, high label cost and the utilization of ubiquitous link information in social media, the unsupervised online feature selection framework—USFS—is proposed and applied into social media in [100]. The USFS is the combination of sparsity-based feature selection and online feature selection described above.

4.3 Multimedia retrieval applications

There is a rapid growth of the amount of multimedia data from real-world multimedia sharing Web sites, such as Flickr and Youtube. It is well known that we can obtain lots of features from multimedia such as images and videos. However, the obtained features are often overcomplete to describe certain semantics. Therefore, the selection of limited discriminative features for certain semantics is hence crucial to make the understanding of multimedia more interpretable [101]. As introduced in [101], many sparsity-based feature selection approaches have been developed in computer vision and multimedia retrieval. Such as Wright [102] transform the face recognition problem into a linear regression problem with sparse constraints for regression coefficients. Since the feature selection methods applied into multimedia have been surveyed in [101], we just emphasize the application of feature selection for "semantic gap" in multimedia retrieval. As we have known the "semantic gap," which is the gap between high-level semantic concepts and low-level visual features, is the fundamental problem in multimedia retrieval, and it is far from being solved. Fortunately, feature selection can alleviate this problem to some extent. For example, in the paper [103], Jiang et al. investigate online feature selection in the relevance feedback learning process to improve the retrieval performance of the region-based image retrieval system. Since the goal of feature selection is to find the optimal feature subspace where the “relevant” and “irrelevant” image sets during a query session can be best separated, the similarity between these two sets is used as the feature selection criterion. They implement an effective online feature selection algorithm in a boosting manner, which incorporates the proposed feature selection criterion with the Real AdaBoost framework [104] to select the optimal features and combine the incrementally learned classifiers over the selected features into a strong ensemble classifier.

Discussion The feature selection algorithms in any topics above have been applied into real-world data to validate their performance; then, feature selection has wide-ranging applications. At the same time, many new applications are emerging, such as software defect prediction [105].

5 Future work and conclusion

In this section, we discuss some future work of feature selection.

First, the aim of feature selection is to choose an optimal feature subset. However, the outputs of some feature selection algorithms are feature ranking (weighting). Then, it still needs to determine the subset from ranking result. If the number of important feature is known, this determination is very easy, we just need to choose the important features one by one from the ranking set until the number of features meets our requirement. Unfortunately, without any prior knowledge, the number of important feature is unknown. Then, the transformation of feature ranking to feature subset is still an open model selection issue in feature selection research.

Second, the research of feature selection closely follows the development of machine learning. When any new machine learning paradigms are emerged, the corresponding feature selection topics also will be studied. For example, adversarial machine learning is the current hot topic, and then, adversarial feature selection is studied soon. On the other hand, different machine learning paradigms are often combined to solve a special problem; then, the combination of corresponding feature selection topics is also an interesting work. For instance, we can study the integration of online feature selection and multi-label feature selection to handle the streaming data with multi-label.

Finally, traditional feature selection algorithms focus on classification or clustering performance; however, some other properties about feature selection should also be paid more attention, such as scalability, stability, and security, especially in big data era. We summarize the different properties with respect to current algorithms for each topic in Table 3, and “Y” means current algorithms have this property.

From the table, we can observe that most algorithms have one characteristic; then, the improvement of current feature selection algorithms to acquire other traits is one of the exciting research directions.

Since deep learning has attracted more attention in feature generation, we also like to discuss the relationship between the deep learning and feature selection. Given large amounts of data, instead of designing a handcraft feature representation, a deep learning algorithm tends to learn a good abstract representation for the current task with a series of nonlinear transformations [106]. So, with the rise of deep learning, it seems that you can do advanced

Table 3 The summary of properties related to current algorithms in each topic

	Scalability	Stability	Security
Topic 1: Stable feature selection		Y	
Topic 2: Sparsity-based feature selection	Y		
Topic 3: Multi-source feature selection	Y		
Topic 4: Distributed feature selection	Y		
Topic 5: Multi-view feature selection	Y		
Topic 6: Multi-label feature selection	Y		
Topic 7: Privacy-preserving feature selection			Y
Topic 8: Adversarial feature selection			Y
Topic 9: Online feature selection	Y		

machine learning without any feature selection. However, in some cases where the number of data points is not sufficiently large, deep learning should be combined with feature selection to obtain better learning performance because with a fixed number of instances, removing irrelevant features is equivalent to exponentially increasing the number of instance. So deep learning is expected to face challenges dealing with HDSSS data. Moreover, we give another example to describe the relationship between deep learning and feature selection. In heterogeneous multi-modal information fusion, each independent modality is characterized by a single feature group, and then, these different modalities are sent to different branches of the multi-modal deep neural networks, yielding refined feature representations with multiple nonlinear transformations based upon the given original modalities. When all the feature groups are transformed by the multi-modal deep neural networks, the outputs of the refined features extracted from the top layer of each branch are concatenated into a new feature vector. Then, a feature selection algorithm is adopted to produce an optimal weight vector for this concatenation. According to this weight vector, the most relevant feature groups with respect to the current task are picked out. Finally, these selected features are used in the final recognition task [107]. On the other hand, feature selection is also an important tool for knowledge discovery and has merits on its own. In this case, our goal is to choose the key features rather than serving as a tool for data preprocessing; then, we prefer feature selection to deep learning and keep the interpretability of original features. For instance, in microarray analysis, biologists are interested in finding a small number of features (genes or proteins) that explain the mechanisms driving different behaviors of microarray samples. A feature selection algorithm is often used to choose a subset of genes or proteins, while deep learning is not suitable.

Feature selection is an ever evolving frontier in data mining, machine learning and statistics. Along with the fast development of machine learning, the scopes of feature selection research and application are also broadened. In this paper, we overview several challenges brought by Big Data, HDSSS problems, multi-label data, privacy preserving, etc. And we selectively discuss some hot topics under these challenges. For each topic, after brief analysis of the existing problem, the current research findings are summarized and followed by a short discussion. We then introduce some current applications of feature selection, such as bioinformatics, social media, and multimedia retrieval. We also discuss some general issues and future work for feature selection. This review is done mostly based on our experience on feature selection more than fifteen years and performing an automated text mining literature analysis on feature selection similar to [108] is also one of our future work.

Acknowledgements This work was partially supported by Natural Science Foundation of China (Nos. 61603197, 91646116), Ministry of Education/China Mobile joint research Grant under Project No. 5-10, Scientific, Technological Support Project (Society) of Jiangsu Province (No. BE2016776), Natural Science Foundation of Jiangsu Province (No. BK20140885), Six talent peaks project in Jiangsu Province under Grant XYDXXJS-CXTD-006 and Jiangsu Qinlan Project.

References

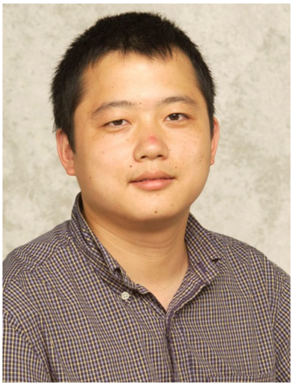
1. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 31:1157–1182
2. Liu H, Yu L (2005) Toward integrating feature selection algorithms for classification and clustering. IEEE Trans Knowl Data Eng 17:494–502
3. Hughes GF (1968) On the mean accuracy of statistical pattern recognizers. IEEE Trans Inf Theory 14:55–63
4. Miller AJ (1984) Selection of subsets of regression variables. J R Stat Soc 147:389–425
5. Blum A, Langle P (1997) Selection of relevant features and examples in machine learning. Artif Intell 97:245–271
6. Kohavi R, John G (1997) Wrappers for feature subset selection. Artif Intell 97:273–324
7. Inza I, Larranaga P, Blanco R, Cerrolaza AJ (2004) Filter versus wrapper gene selection approaches in DNA microarray domains. Artif Intell Med 31:91–103
8. Forman G (2003) An extensive empirical study of feature selection metrics for text classification. J Mach Learn Res 3:1289–1305
9. Blum AL, Rivest RL (1992) Training a 3-node neural networks is NP-complete. Neural Netw 5:117–127
10. Frank A, Asuncion A (2010) UCI machine learning repository. http://archive.ics.uci.edu/ml
11. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286:531–537
12. Singh D, Febbo PG, Ross K (2002) Gene expression correlates of clinical prostate cancer behavior. Cancer Cell 2:203–209
13. Bhattacharjee A, Richards WG, Staunton J, Li C, Monti S (2001) Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. Proc Natl Acad Sci USA 98:13790–13795
14. Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon cancer tissues probed by oligonucleotide arrays. Proc Natl Acad Sci USA 96:6745–6750
15. Zhao Z (2010) Spectral feature selection for mining ultrahigh dimensional data, Ph.D. thesis. Arizona State University
16. Guyon I, Gunn S, Nikravesh M, Zadeh L (2006) Feature extraction, foundations and applications. Springer, Physica-Verlag, New York
17. Dash M, Liu H (1997) Feature selection for classification. Intell Data Anal 1:131–156
18. Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Comput Electr Eng 40:16–28
19. Tang JL, Alelyani S, Liu H (2014) Feature selection for classification—a review. In: Aggarwal C (ed) Data classification: algorithms and applications. CRC Press, Boca Raton
20. Li JD, Cheng KW, Wang SH, Morstatter F, Trevino RP, Tang JL, Liu H (2016) Feature selection: a data perspective, vol 3, pp 1–73. arXiv:1601.07996
21. Peng HC, Long FH, Ding C (2005) Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 27:1226–1238
22. Mitra P, Murthy CA, Pal SK (2002) Unsupervised feature selection using feature similarity. IEEE Trans Pattern Anal Mach Intell 24:301–312
23. Hall MA (2000) Correlation-based feature selection for discrete and numeric class machine learning. In: Proceedings of international conference on machine learning, pp 359–366
24. Yu L, Liu H (2003) Feature selection for high-dimensional data: a fast correlation-based filter solution. In: Proceedings of international conference on machine learning, pp 856–863
25. Yu L, Liu H (2004) Efficient feature selection via analysis of relevance and redundancy. J Mach Learn Res 5:1205–1224
26. Saeys Y, Abeel T, de Peer YV (2008) Robust feature selection using ensemble feature selection techniques. In: Proceedings of the 25th European conference on machine learning and knowledge discovery in databases, Banff, pp 313–325

27. Han Y, Yu L (2010) A variance reduction framework for stable feature selection. In: Proceedings of the international conference on data mining, pp 206–215
28. Loscalzo S, Yu L, Ding C (2009) Consensus group stable feature selection. In: Proceedings of ACM SIGKDD conference on knowledge discovery and data mining, pp 567–575
29. Abeel T, Helleputte T, de Peer YV, Dupont P, Saeys Y (2010) Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. Bioinformatics 26:392–398
30. Li Y, Gao SY, Chen SC (2012) Ensemble feature weighting based on local learning and diversity. In: AAAI Conference on artificial intelligence, pp 1019–1025
31. Woznica A, Nguyen P, Kalousis A (2012) Model mining for robust feature selection. In: Proceedings of ACM SIGKDD conference on knowledge discovery and data mining, pp 913–921
32. Yu L, Han Y, Berens ME (2012) Stable gene selection from microarray data via sample weighting. IEEE/ACM Trans Comput Biol Bioinform 9:262–272
33. Yu L, Ding C, Loscalzo S (2008) Stable feature selection via dense feature groups. In: Proceedings of ACM SIGKDD conference on knowledge discovery and data mining, pp 803–811
34. He ZY, Yu WC (2010) Stable feature selection for biomarker discovery. Comput Biol Chem 34:215–225
35. Li Y, Huang SS, Chen SC, Si J (2013) Stable l2-regularized ensemble feature weighting. In: Proceedings of the 11th international workshop on multiple classifier systems, pp 167–178
36. Li Y, Si J, Zhou GJ, Huang SS, Chen SC (2015) Frel: a stable feature selection algorithm. IEEE Trans Neural Netw Learn Syst 26:1388–1402
37. Crammer K, Bachrach RG, Navot A, Tishby N (2002) Margin analysis of the LVQ algorithm. In: Proceedings of advances in neural information processing systems, pp 462–469
38. Strehl A, Ghosh J (2002) Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J Mach Learn Res 3:583–617
39. Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Stat Soc Ser B (Stat Methodol) 58:267–288
40. Ng AY (2004) Feature selection, l1 vs. l2 regularization, and rotational invariance. In: Proceedings of international conference on machine learning, pp 78–85
41. Jenatton R, Obozinski G, Bach F (2010) Structured sparse principal component analysis. In: Proceedings of international conference on artificial intelligence and statistics
42. Yuan M, Lin Y (2006) Model selection and estimation in regression with grouped variables. J R Stat Soc Ser B (Stat Methodol) 68:49–67
43. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J R Stat Soc Ser B (Stat Methodol) 67:301–320
44. Kim S, Xing EP (2010) Tree-guided group lasso for multi-task regression with structured sparsity. In: Proceedings of the 27th international conference on machine learning
45. Wang J, Zhou JY, Liu J, Wonka P, Ye JP (2014) A safe screening rule for sparse logistic regression. In: Proceedings of advances in neural information processing systems, pp 1053–1061
46. Wang J, Ye JP (2015) Safe screening for multi-task feature learning with multiple data matrices. In: Proceedings of the 32nd international conference on machine learning
47. Zhao Z, Wang JX, Sharma S, Agarwal N, Liu H, Chang Y (2010) An integrative approach to identifying biologically relevant genes. In: Proceedings of SIAM International conference on data mining
48. Weinberger K, Dasgupta A, Langford J, Smola A, Attenberg J (2009) Feature hashing for large scale multitask learning. In: Proceedings of international conference on machine learning
49. Chu CT, Kim SK, Lin YA, Yu YY, Bradski G, Ng A, Olukotun K (2007) Map-reduce for machine learning on multicore. In: Proceedings of advances in neural information processing systems
50. Snir M, Otto S, Lederman SH, Walker D, Dongarra J (1995) MPI: the complete reference, 1st edn. MIT Press, Cambridge
51. Dean J, Ghemawat S (2008) Mapreduce: simplified data processing on large clusters. Commun ACM 51:107–113
52. Zhao ZA, Liu H (2012) Spectral feature selection for data mining. Taylor and Francis Group, London
53. Zhao Z, Zhang RW, Cox J, Duling D, Sarle W (2013) Massively parallel feature selection: an approach based on variance preservation. Mach Learn 92:195–220
54. Das K, Bhaduri K (2010) H. Kargupta: A local asynchronous distributed privacy preserving feature selection algorithm for large peer-to-peer networks. Knowl. Inf Syst 24:341–367
55. Wu X, Zhu X, Wu GQ, Ding W (2014) Data mining with big data. IEEE Trans Knowl Data Eng 26:97–107
56. Cao B, He LF, Kong XN, Yu PS, Hao ZF, Ragin AB (2014) Tensor-based multi-view feature selection with applications to brain diseases. In: Proceedings of the 2014 international conference on data mining, pp 40–49

57. Smalter A, Huan J, Lushington G (2009) Feature selection in the tensor product feature space. In: Proceedings of the 2009 international conference on data mining, pp 1004–1009
58. Tang JL, Hu X, Gao HJ, Liu H (2013) Unsupervised feature selection for multi-view data in social media. In: Proceedings of the 2013 SIAM conference on data mining
59. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46:389–422
60. Fang Z, Zhang ZM (2013) Discriminative feature selection for multi-view cross-domain learning. In: Proceedings of ACM international conference of information and knowledge management, pp 1321–1330
61. Chen WZ, Yan J, Zhang BY, Chen Z, Yang Q (2007) Document transformation for multi-label feature selection in text categorization. In: Proceedings of the 7th IEEE conference on data mining, pp 451–456
62. Quinlan JR (1986) Induction of decision trees. Mach Learn 1:81–106
63. Kass GV (1980) An exploratory technique for investigating large quantities of categorical data. Appl Stat 119–127
64. Yan J, Liu N, Zhang B, Yan S, Chen Z, Cheng Q, Fan W, Ma WY (2005) OCFS: optimal orthogonal centroid feature selection for text categorization. In: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval, pp 122–129
65. Lastra G, Luaces O, Quevedo JR, Bahamonde A (2011) Graphical feature selection for multilabel classification tasks. In: Proceedings of the 10th international conference on advances in intelligent data analysis, pp 281–305
66. Kong X, Yu PS (2012) gMLC: a multi-label feature selection framework for graph classification. Knowl Inf Syst 31:281–305
67. Gu QQ, Li ZH, Han JW (2011) Correlated multi-label feature selection. In: Proceedings of the 20th ACM international conference on information and knowledge management, pp 1087–1096
68. Elisseeff A, Weston J (2001) A kernel method for multi-labelled classification. In: Advances in neural information processing systems, pp 681–687
69. Yan P, Li Y (2016) Graph-margin based multi-label feature selection. In: European conference on machine learning, pp 540–555
70. Perkins S, Theiler J (2003) Online feature selection using grafting. In: Proceedings of international conference on machine learning, pp 592–599
71. Wu X, Yu K, Wang H, Ding W (2010) Online streaming feature selection. In: Proceedings of international conference on machine learning, pp 1159–1166
72. Zhou D, Huang J, Scholkopf B (2005) Learning from labeled and unlabeled data on a directed graph. In: Proceedings of international conference on machine learning, pp 1036–1043
73. Yu K, Wu XD, Ding W, Pei J (2014) Towards scalable and accurate online feature selection for big data. In: Proceedings of IEEE conference on data mining, pp 660–669
74. Sengupta D, Bandyopadhyay S, Sinha D (2017) A scoring scheme for online feature selection: simulating model performance without retraining. IEEE Trans Neural Netw Learn Syst 28:405–414
75. Wang J, Zhao ZQ, Hu XG, Cheung YM, Wang M, Wu XD (2013) Online group feature selection. In: Proceedings of international joint conference on artificial intelligence
76. Wang J, Zhao P, Hoi S, Jin R (2014) Online feature selection and its applications. IEEE Trans Knowl Data Eng 26:698–710
77. Zhang Q, Zhang P, Long G, Ding W, Zhang C, Wu X (2015) Towards mining trapezoidal data streams. In: Proceedings of IEEE international conference on data mining, pp 1111–1116
78. Avidan S, Butman M (2006) Efficient methods for privacy preserving face detection. In: Advances in neural information processing systems, pp 57–64
79. Friedman J, Hastie T, Tibshirani R (2000) Additive logistic regression: a statistical view of boosting. Ann Stat 28:337–407
80. Zhou Q, Zhou H, Li T (2016) Cost-sensitive feature selection using random forest: selecting low-cost subsets of informative features. Knowl-Based Syst 95:1–11
81. Dwork C (2006) Differential privacy. In: Proceedings of international colloquium on automata, languages and programming, pp 1–12
82. Yang J, Li Y (2014) Differential privacy feature selection. In: Proceedings of international joint conference on neural networks, pp 4182–4189
83. Li Y, Yang J, Ji W (2016) Local learning-based feature weighting with privacy preservation. Neurocomputing 174:1107–1115
84. Sun YJ, Todorovic S, Goodison S (2010) Local learning based feature selection for high dimensional data analysis. IEEE Trans Pattern Anal Mach Intell 32:1–18
85. Barreno M, Nelson B, Joseph AD, Tygar JD (2010) The security of machine learning. Mach Learn 81:121–148

86. Huang L, Joseph AD, Nelson B, Rubinstein BIP, Tygar JD (2011) Adversarial machine learning. In: Proceedings of 4th ACM workshop on artificial intelligence and security, pp 43–58
87. Biggio B, Fumera G, Roli F (2014) Security evaluation of pattern classifiers under attack. IEEE Trans Knowl Data Eng 26:984–996
88. Li B, Vorobeychik Y (2014) Feature cross-substitution in adversarial classification. In: Proceedings of advances in neural information processing systems, pp 2087–2095
89. Xiao H, Biggio B, Brown G, Fumera G, Eckert C, Roli F (2015) Is feature selection secure against training data poisoning? In: Proceedings of the 32th international conference on machine learning
90. Zhang F, Chan PPK, Biggio B, Yeung DS, Roli F (2015) Adversarial feature selection against evasion attacks. IEEE Trans Cybern 46:766–777
91. Saeys Y, Inza I, Larranaga P (2007) A review of feature selection techniques in bioinformatics. Bioinformatics 23:2507–2517
92. Bolon-Canedo V, Sanchez-Marono N, Alonso-Betanzos A, Benitez JM, Herrera F (2014) A review of microarray datasets and applied feature selection methods. Inf Sci 282:111–135
93. Nie FP, Huang H, Cai X, Ding C (2010) Efficient and robust feature selection via joint l21-norms minimization. Adv Neural Inf Process Syst 23:1813–1821
94. Tang JL, Liu H (2012) Feature selection with linked data in social media. In: SIAM international conference on data mining
95. Tang JL, Liu H (2012) Unsupervised feature selection for linked social media data. In: Eighteenth ACM SIGKDD international conference on knowledge discovery and data mining
96. Tang JL, Liu H (2014) Feature selection for social media data. ACM Trans Knowl Discov Data 8:1–27
97. Tang JL, Liu H (2014) An unsupervised feature selection framework for social media data. IEEE Trans Knowl Data Eng 26:2914–2927
98. Newman MEJ, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E 69:026113-1-026113-15
99. Luxburg U (2007) A tutorial on spectral clustering. Stat Comput 17:395–416
100. Li JD, Tang JL, Hu X, Liu H (2015) Unsupervised streaming feature selection in social media. In: Proceedings of ACM international conference of information and knowledge management
101. Wu F, Han YH, Liu X, Shao J, Zhuang YT, Zhang ZF (2012) The heterogeneous feature selection with structural sparsity for multimedia annotation and hashing: a survey. Int J Multimed Inf Retr 1:3–15
102. Wright J, Yang A, Ganesh A, Sastry S, Ma Y (2009) Robust face recognition via sparse representation. IEEE Trans Pattern Anal Mach Intell 31:210–227
103. Jiang W, Er GH, Dai QH, Gu JW (2006) Similarity-based online feature selection in content-based image retrieval. IEEE Trans Image Process 15:702–712
104. Friedman J, Hastie T, Tibshirani R (2000) Additive logistic regression: a statistical view of boosting. Ann Stat 38:337–374
105. Khoshgoftaar TM, Gao KH, Napolitano A, Wald R (2014) A comparative study of iterative and non-iterative feature selection techniques for software defect prediction. Info Syst Frontiers 16:801–822
106. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313:504–507
107. Zhao L, Hu Q, Wang W (2015) Heterogeneous feature selection with multi-modal deep neural networks and sparse group lasso. IEEE Trans Multimed 17:1936–1948
108. Moro S, Cortez P, Rita P (2015) Business intelligence in banking: a literature analysis from 2002 to 2013 using text mining and latent Dirichlet allocation. Expert Syst Appl 42:1314–1324

Yun Li received the Ph.D. degree in Computer Science from Chongqing University, Chongqing, China. He is a professor in the College of Computer Science, Nanjing University of Posts and Telecommunications, China. Prior to that, he was the postdoctoral fellow in Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His research mainly focuses on machine learning, data mining and parallel computing. He has published more than 40 refereed research papers. His research is currently sponsored by the National Natural Science Foundation of China (NSFC) and Natural Science Foundation of Jiangsu Province.



Tao Li received the Ph.D. degree in Computer Science from the Department of Computer Science, University of Rochester, Rochester, NY, in 2004. He is currently a professor with the School of Computer Science, Nanjing University of Posts and Telecommunications (NJUPT). He is also a professor with the School of Computing and Information Sciences, Florida International University, Miami, FL. His research interests include data mining, computing system management, information retrieval, and machine learning. He received the US National Science Foundation (NSF) CAREER Award and multiple IBM Faculty Research Awards.

Huan Liu received the B.Eng. degree in Computer Science and Electrical Engineering from Shanghai JiaoTong University, Shanghai, China, and the Ph.D. degree in Computer Science from the University of Southern California, Los Angeles, California. He is currently a professor of computer science and engineering with Arizona State University (ASU), Tempe, Arizona. Before he joined ASU, he worked in Telecom Australia Research Labs and was on the faculty with the National University of Singapore. He was recognized for excellence in teaching and research in computer science and engineering with Arizona State University. His research interests include data mining, machine learning, social computing, and artificial intelligence, investigating problems that arise in many real-world, data-intensive applications with high-dimensional data of disparate forms such as social media. His well-cited publications include books, book chapters, encyclopedia entries as well as conference and journal papers. He serves on journal editorial boards and numerous conference program committees, and is a founding organizer of the International Conference Series on Social Computing, Behavioral-Cultural Modeling, and Prediction (http://sbp.asu.edu/). He is a fellow of the IEEE and an ACM distinguished scientist.